MAP-i Doctoral Program

PhD thesis proposal

Title — WIDENING THE SCOPE OF APPLICATION OF THE SPIRAL PARALLEL CODE GENERATING SYSTEM

Context — The wide spread of multi-core CPU architectures has brought parallelism and concurrency onto everybody's working desk. This poses a major challenge to the average software developer, who very likely has been trained to "think sequentially". Yet some programming paradigms (such as the functional one) enable one to "think parallel" [2] by abstracting from many (microscopic, "semicolon oriented") programming details. Programming in such a declarative style is governed by a rich algebra of programming which makes it possible to calculate parallel, efficient versions of algorithms which have been specified sequentially. This is so because declarative programming very naturally expresses algorithms in terms of "divide and conquer" strategies. The main lesson learnt is that mathematics embodies parallel evaluation in a very natural way.

Such is the principle behind Spiral [5], a program generation system developed at CMU (ESE) which performs optimizations automatically to produce code that rivals the best human-tuned code in performance. It includes various optimizations such as loop unrolling, adaptation to the memory hierarchy, and the use of special instruction sets. Spiral is focused on optimization for digital signal processing (DSP) algorithms. It entirely autonomously generates platform-tuned implementations of signal processing transform such as the discrete Fourier transform, discrete cosine transform, and many others.

Aims — Spiral is essentially a term rewriting system which works at a very high level of abstraction. It inputs formulæ specifying transform algorithms and outputs equivalent "vectorized" formulæ which are directly mapped into short vector code. It doing this it shares the core principle of formal program development: build a mathematical *model* first, then *transform* it (calculate) so as to reach efficient implementations.

The main aim of this project is to bring the two cultures together, so as to broaden the scope of application of parallel code generating systems such as Spiral. In theory, every DSL based on combinators whose semantics are parallelizable by calculation [3] should be under the scope of tools such as Spiral. A number of recent results on calculating parallelism (see eg. [4]) should be taken into account. The use of so-called strategic term rewriting [1] techniques, which generalize the traditional term rewriting paradigm by making rewriting strategies fully typed and programmable, will be considered. The likely application of the results of the project to widespread data processing languages such as eg. SQL is also envisaged.

Contact — José N. Oliveira (CCTC) and Markus Püschel (ESE, CMU)

R&D Unit — CCTC (`http://cctc.di.uminho.pt/`) and ECE, CMU (`http://www.ece.cmu.edu/`). Negotiations with the CMU-Portugal Program are under way.

References

[1] T.L. Alves, P.F. Silva, J. Visser, and J.N. Oliveira. Strategic term rewriting and its application to a VDM-SL to SQL conversion. In *FM 2005*, volume 3582 of *LNCS*, pages 399–414. Springer-Verlag, 2005.

[2] J. Backus. Can programming be liberated from the von Neumann style? a functional style and its algebra of programs. *CACM*, 21(8):613–639, August 1978.

[3] Zhenjiang Hu, Masato Takeichi, and Wei-Ngan Chin. Parallelization in calculational forms. In *POPL*, pages 316–328, 1998.

[4] Kazutaka Morita, Akimasa Morihata, Kiminori Matsuzaki, Zhenjiang Hu, and Masato Takeichi. Automatic inversion generates divide-and-conquer parallel programs. In *PLDI*, pages 146–155, 2007.

[5] Markus Püschel, José M. F. Moura, Jeremy Johnson, David Padua, Manuela Veloso, Bryan Singer, Jianxin Xiong, Franz Franchetti, Aca Gacic, Yevgen Voronenko, Kang Chen, Robert W. Johnson, and Nick Rizzolo. SPIRAL: Code generation for DSP transforms. Proceedings of the IEEE special issue on "Program Generation, Optimization, and Adaptation," Vol. 93, No. 2, 2005, pp. 232-275.